

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平11-66043

(43) 公開日 平成11年(1999) 3月9日

(51) Int.Cl.^{*}
G 0 6 F 17/14
17/11
G 0 6 T 1/00
7/20

識別記号

F I
G 0 6 F 15/332
15/32
15/62
15/70
S
3 8 0
4 1 0

審査請求 未請求 請求項の数22 O L (全 26 頁)

(21) 出願番号 特願平9-230577

(22) 出願日 平成9年(1997) 8月27日

(71) 出願人 000005223

富士通株式会社

神奈川県川崎市中原区上小田中4丁目1番
1号

(72) 発明者 前田 芳晴

神奈川県川崎市中原区上小田中4丁目1番
1号 富士通株式会社内

(72) 発明者 エティエンヌ パーデット

カナダ, ビー. シー. ブイ5ビー 2ビ
ー1 パーナビー, カーティス ストリー
ト 6770

(74) 代理人 弁理士 大菅 義之 (外1名)

最終頁に続く

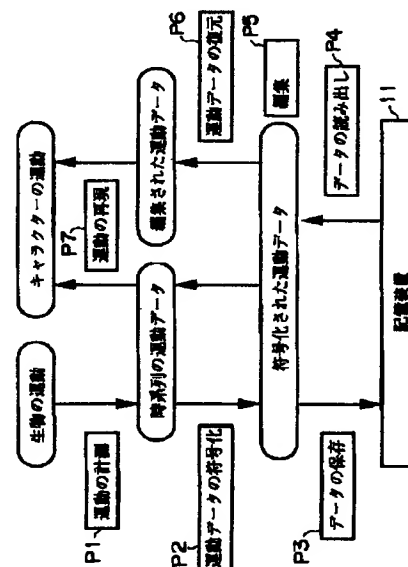
(54) 【発明の名称】 時系列データの符号化と編集を行うデータ変換装置および方法

(57) 【要約】

【課題】 キャラクターの運動を表示するシステムにおいて、運動データを効率良く保存し、その再利用性を向上させることが課題である。

【解決手段】 生物の運動を計測して、時系列の運動データを収集し (P1)、それを符号化して (P2)、記憶装置11に保存する (P3)。次に、符号化された運動データを読み出して (P4)、それを編集し、必要なデータを作成する (P5)。また、符号化された運動データから運動データを復元し (P6)、それに基づいてキャラクターの運動を再現する (P7)。

運動データの処理過程を示す図



1

【特許請求の範囲】

【請求項 1】 任意の物体の運動を計測して得られた時系列の運動データを保存し、該運動データを用いた情報処理を行うデータ変換装置であって、

前記運動データを任意の基底関数の重み付き加算で近似表現し、該重み付き加算の重み係数の配列を生成する近似手段と、

前記重み係数の配列を前記運動データに対応する符号データとして格納する格納手段とを備えることを特徴とする、コンピュータを用いたデータ変換装置。

【請求項 2】 前記近似手段は、前記基底関数として、滑らかで局所的な関数を用いることを特徴とする請求項 1 記載のデータ変換装置。

【請求項 3】 前記近似手段は、一定時間内に計測された前記運動データを行列データとして表現し、該行列データを前記重み係数の配列と前記基底関数の値を要素とする行列との積の形式で表現した方程式を解いて、該重み係数の配列を求めることを特徴とする請求項 2 記載のデータ変換装置。

【請求項 4】 前記近似手段は、前記基底関数として B-スプライン関数を用い、前記一定時間内の B-スプライン関数の値を用いて前記基底関数の行列を表現することを特徴とする請求項 3 記載のデータ変換装置。

【請求項 5】 前記近似手段は、疑似逆行列を用いた方法とガウス消去法のうちのいずれかを用いて、前記方程式を解くことを特徴とする請求項 3 記載のデータ変換装置。

【請求項 6】 任意の物体の運動を計測して得られた時系列の運動データを用いて、仮想的なキャラクターの画像を出力するための情報処理を行うデータ変換装置であって、

前記運動データを符号化して符号データを生成する符号化手段と、

前記符号データを格納する格納手段と、

前記格納手段から前記符号データを取り出し、該符号データを編集して、新たな符号データを生成する編集手段と、

前記新たな符号データに対応する運動データを復元する復元手段と、

復元された運動データを用いて前記キャラクターの画像を生成し、出力する出力手段とを備えることを特徴とする、コンピュータを用いたデータ変換装置。

【請求項 7】 前記符号化手段は、前記運動データを任意の基底関数の重み付き加算で近似表現し、該重み付き加算の重み係数の配列を生成して、該重み係数の配列を前記符号データとして前記格納手段に格納することを特徴とする請求項 6 記載のデータ変換装置。

【請求項 8】 前記編集手段は、第 1 の座標系で記述された前記符号データを第 2 の座標系で記述された符号データに変換する座標変換手段を含むことを特徴とする請

2

求項 6 記載のデータ変換装置。

【請求項 9】 前記編集手段は、運動の開始時刻を遅らせるデータ変換と、運動時間を伸張するデータ変換と、運動時間を短縮するデータ変換のうち、少なくとも一つの変換を行う時間変換手段を含むことを特徴とする請求項 6 記載のデータ変換装置。

【請求項 10】 前記編集手段は、前記符号データの平行移動と回転とサイズ変更のうち少なくとも一つのデータ変換を行う空間変換手段を含むことを特徴とする請求項 6 記載のデータ変換装置。

【請求項 11】 前記編集手段は、前記符号データの分離と連結と融合のうち少なくとも一つのデータ変換を行う変換手段を含むことを特徴とする請求項 6 記載のデータ変換装置。

【請求項 12】 前記編集手段は、前記符号データに含まれる任意の重み係数を変更する係数調節手段を含むことを特徴とする請求項 6 記載のデータ変換装置。

【請求項 13】 前記編集手段は、逆ダイナミクスを利用して、運動を表現する符号データを生物の関節トルクを表現する符号データに変換するデータ変換と、順ダイナミクスを利用して、生物の関節トルクを表現する符号データを運動を表現する符号データに変換するデータ変換のうち、少なくとも一つのデータ変換を行う変換手段を含むことを特徴とする請求項 6 記載のデータ変換装置。

【請求項 14】 前記符号化手段による符号化処理と前記編集手段による編集処理のうち少なくとも一方を、グラフィカル・ユーザ・インタフェースを介してユーザとの対話形式で行うインタフェース手段をさらに備えることを特徴とする請求項 6 記載のデータ変換装置。

【請求項 15】 任意の物体の運動を計測して得られた時系列の運動データを用いて、仮想的なキャラクターの画像を出力するための情報処理を行うデータ変換装置であって、

前記運動データを圧縮して圧縮データを生成する圧縮手段と、

前記圧縮データを格納する格納手段と、

前記格納手段から前記圧縮データを取り出し、該圧縮データを編集して、新たな圧縮データを生成する編集手段と、

前記新たな圧縮データに対応する運動データを復元する復元手段と、

復元された運動データを用いて前記キャラクターの画像を生成し、出力する出力手段とを備えることを特徴とする、コンピュータを用いたデータ変換装置。

【請求項 16】 任意の物体の運動を計測して得られた時系列の運動データを保存し、該運動データを用いた情報処理を行うデータ変換装置であって、

前記運動データに対応する符号データを格納する格納手段と、

3

前記格納手段から前記符号データを取り出し、該符号データを重み係数の配列として用いて任意の基底関数の重み付き加算を計算し、前記運動データを近似的に復元する復元手段と、

復元された運動データを用いて画像を生成し、出力する出力手段とを備えることを特徴とする、コンピュータを用いたデータ変換装置。

【請求項 17】 任意の物体の運動を計測して得られた時系列の運動データを保存し、該運動データを用いた情報処理を行うコンピュータのためのプログラムを記録した記録媒体であって、

前記運動データを任意の基底関数の重み付き加算で近似表現し、該重み付き加算の重み係数の配列を生成する機能と、

前記重み係数の配列を前記運動データに対応する符号データとして格納する機能とを前記コンピュータに実現させるためのプログラムを記録したコンピュータ読み取り可能な記録媒体。

【請求項 18】 任意の物体の運動を計測して得られた時系列の運動データを用いて、仮想的なキャラクターの画像を出力するための情報処理を行うコンピュータのためのプログラムを記録した記録媒体であって、

前記運動データを符号化して符号データを生成する機能と、

前記符号データを格納する機能と、

前記符号データを編集して、新たな符号データを生成する機能と、

前記新たな符号データに対応する運動データを復元する機能と、

復元された運動データを用いて前記キャラクターの画像を生成し、出力する機能とを前記コンピュータに実現させるためのプログラムを記録したコンピュータ読み取り可能な記録媒体。

【請求項 19】 任意の物体の運動を計測して得られた時系列の運動データを保存し、該運動データを用いた情報処理を行うコンピュータのためのプログラムを記録した記録媒体であって、

前記運動データに対応する符号データを格納する機能と、

前記格納手段から前記符号データを取り出し、該符号データを重み係数の配列として用いて任意の基底関数の重み付き加算を計算し、前記運動データを近似的に復元する機能と、

復元された運動データを用いて画像を生成し、出力する機能とを前記コンピュータに実現させるためのプログラムを記録したコンピュータ読み取り可能な記録媒体。

【請求項 20】 任意の物体の運動を計測して、時系列の運動データを取得し、

前記運動データを任意の基底関数の重み付き加算で近似表現して、該重み付き加算の重み係数の配列を生成し、

4

前記重み係数の配列を前記運動データに対応する符号データとして保存し、

前記符号データを用いた情報処理を行うことを特徴とするデータ変換方法。

【請求項 21】 任意の物体の運動を計測して、時系列の運動データを取得し、

前記運動データを符号化して符号データを生成し、

前記符号データを編集して、新たな符号データを生成し、

10 前記新たな符号データに対応する運動データを復元し、復元された運動データを用いて仮想的なキャラクターの画像を生成し、画面表示することを特徴とするデータ変換方法。

【請求項 22】 任意の物体の運動を計測して、時系列の運動データを取得し、

前記運動データに対応する符号データを保存し、

前記格納手段から前記符号データを取り出し、該符号データを重み係数の配列として用いて任意の基底関数の重み付き加算を計算して、前記運動データを近似的に復元し、

20 復元された運動データを用いて画像を生成し、画面表示することを特徴とするデータ変換方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、物体の運動データのような時系列データを符号化して効率的に保存し、符号化された時系列データを編集して再利用するデータ変換装置およびその方法に関する。

【0002】

30 【従来の技術とその問題点】近年、コンピュータ・グラフィックス（CG）やビデオゲーム、および、アニメーションや映画などの分野で、しばしば、人型や動物型などの仮想的キャラクター（以下、単にキャラクターと記す）が用いられている。そして、画面上に表示されたキャラクターを自由自在に動かし、“走る”とか“立ち上がる”など様々な動作を表現することが要望されている。

40 【0003】このような要望に対して、コンピュータを利用してキャラクターの身体の運動データを生成し編集する技術が開発されてきている。ここで、運動データとは、例えば、モーション・キャプチャー・システムなどの運動計測技術を用いて計測された、身体の関節角度や手足・頭の位置などの時系列データに相当する。

【0004】コンピュータを利用して画面上のキャラクターの動きを生成する方法には、大きく分けて、次の3つの方法が考えられる。

（1）生物らしい自然な動きを生成するアルゴリズムを開発し、これによってキャラクター・モデルの動きを生成する方法

50 （2）従来のアニメーション作成過程にコンピュータを

5

利用し、CGで適当な時間間隔毎の静止画を作成したり、中割りを自動生成したりする方法

(3) 人間などの生物の実際の動きを計測装置によって運動データとして取り込み、3次元(3D)キャラクター・モデルに計測した運動データを再現(playback)させる方法

第1の方法は、生物らしい自然な動きを生成するアルゴリズムが開発できれば、多種多様な運動を自由自在に作成できる。また、キャラクターの身体サイズなどの物理特性の変化にも対応できるなどの非常に優れた利点を持っている。しかし、有効なアルゴリズムを開発すること自体が極めて困難であり、さらにキャラクターの精密なダイナミクス・モデルを必要とするなどの問題もある。このため、現状では限定された運動に対してアルゴリズムやダイナミクス・モデルが開発されているのみであり、多種多様な運動を生成できるようになるにはまだまだ研究が必要である。

【0005】また、第2の方法は、現在もっとも利用されている方法である。この方法では、動作の種類毎に膨大な数の静止画が必要であり、多くの時間と労力を要する。また、生物らしい自然な動きを表現できるかどうかは、画像製作者の能力に依存している。

【0006】一方、第3の方法は、生物の実際の運動を計測して、その結果をコンピュータを利用してプレイバックするので、キャラクターに生物らしい自然な動きを行わせることができる。しかし、この方法では、例えば、“歩く”とか“走る”などの動作毎に、実際の動作を計測して記録することが必要になる。なぜなら、記録していない動作は再現できないからである。

【0007】したがって、キャラクターに様々な動作を行わせる場合には、あらかじめ多数の運動データを計測し、記録しておかなければならず、それらを保存しておくために膨大な記憶容量が必要となる。さらに、計測した運動データは、実際に動作を行った計測対象の身体サイズや状況に依存するので、キャラクターや状況が変われば運動データを再度計測し直すことが必要になる。このように、第3の方法では、運動データが膨大な記憶容量を必要とすること、および、運動データの再利用性が低いことが問題点である。

【0008】本発明の課題は、上記のような第3の方法を利用したプレイバック技術において、運動データを効率良く保存し、その再利用性を向上させるデータ変換装置およびその方法を提供することである。

【0009】

【課題を解決するための手段】図1は、本発明のデータ変換装置の原理図である。図1のデータ変換装置は、本発明の第1および第2の原理のデータ変換装置を含む。

【0010】第1の原理のデータ変換装置は、コンピュータを用いて実現され、近似手段1および格納手段2を備える。そして、任意の生物を含む任意の物体の運動を

6

計測して得られた時系列の運動データを保存し、その運動データを用いた情報処理を行う。

【0011】近似手段1は、上記運動データを任意の基底関数の重み付き加算で近似表現し、その重み付き加算の重み係数の配列を生成する。格納手段2は、その重み係数の配列を上記運動データに対応する符号データとして格納する。

【0012】近似手段1は、例えば、B-スプライン関数のような滑らかで局所的な基底関数を用いて、その重み付き加算により得られる曲線で運動データを近似的に表現する。そして、各基底関数の重み係数を、運動データを表現する離散的な符号として、格納手段2に格納する。

【0013】格納手段2に保存された符号データを重み係数として、基底関数の重み付き加算を行えば、元の運動データを容易に復元することができる。基底関数の関数形はあらかじめ決まっており、入力される運動データの形状によって変化することはない。このため、必要とされる一組の基底関数を表すデータを保持しておけば、それを任意の運動データの復元に用いることができる。

【0014】また、重み係数の配列データは、一般に、計測された運動データよりはるかに少ないデータ量で表現することができる。したがって、運動データをそのまま保存する場合より、記憶容量を大幅に削減することが可能になる。

【0015】また、第2の原理のデータ変換装置は、コンピュータを用いて実現され、格納手段2、符号化手段3、編集手段4、復元手段5、および出力手段6を備える。そして、任意の物体の運動を計測して得られた時系列の運動データを用いて、仮想的なキャラクターの画像を出力するための情報処理を行う。

【0016】符号化手段3は、上記運動データを符号化して符号データを生成し、格納手段2は、その符号データを格納する。編集手段4は、格納手段2からその符号データを取り出して、それを編集し、新たな符号データを生成する。復元手段5は、新たな符号データに対応する運動データを復元し、出力手段6は、復元された運動データを用いて上記キャラクターの画像を生成し、それを出力する。

【0017】符号化手段3は、例えば、上述した第1の原理の近似手段1と同様の方法で運動データを符号化し、データ量を削減して、格納手段2に格納する。編集手段4は、符号データに対して、座標変換、時間に関する変換、空間に関する変換、分割、連結、融合、手動による重み係数の調節、ダイナミクスを利用した変換などの編集処理を施すことで、符号データを変更する。これにより、元の運動データと異なる運動を表現する新たな符号データが生成される。

【0018】復元手段5は、こうして生成された新たな符号データに、符号化処理と逆の処理を施して、対応す

7

る新たな運動データを生成する。そして、出力手段 6 は、その運動データに基づいて、キャラクターの運動を表現する動画を生成し、それを出力する。これにより、入力された運動データとは異なる運動をキャラクターに行わせることが可能になり、運動データの再利用性が向上する。

【0019】例えば、図 1 の近似手段 1、符号化手段 3、編集手段 4、復元手段 5 は、後述する図 7 の CPU (中央処理装置) 21 とメモリ 22 に対応し、格納手段 2 は記憶装置 25 に対応し、出力手段 6 はディスプレイ装置 24' に対応する。

【0020】

【発明の実施の形態】以下、図面を参照しながら、本発明の実施の形態を詳細に説明する。本発明では、運動データを効率良く保存するために符号化を行い、その再利用性を高めるために、符号化された運動データ (符号化運動データ) の編集を行う。ここで、符号化運動データとは、時系列データの一つである運動データを、本発明の方法によって符号化した結果の符号データを指す。

【0021】図 2 は、本発明の実施形態のデータ変換装置における運動の計測から再現までの処理過程を示している。このデータ変換装置は、人間などの生物の実際の動きを計測装置によって取り込み、ディスプレイ画面上のキャラクター・モデルにそれを再現させる。この処理過程には、運動の計測 (P1)、運動データの符号化 (P2)、符号化運動データの保存 (P3)、符号化運動データの読み出し (P4)、符号化運動データの編集 (P5)、符号化運動データから運動データの復元 (P6)、およびキャラクターの運動の再現 (P7) の情報処理が含まれる。

【0022】本発明は、これらの処理のうち、特に、運動データの符号化、符号化運動データの編集、および運動データの復元に特徴を有する。処理 P2 の符号化では、運動データを効率良く保存して膨大な記憶容量の問題を解決するため、運動データを符号化する。また、処理 P5 の編集では、運動データの再利用性を高めるために、符号化運動データに対して編集を行う。この編集処理では、運動データの座標変換、連結、分割、変形、移動などを可能にする。また、処理 P6 の復元では、処理 P2 と逆の処理を行って運動データを復元する。

【0023】また、他の処理 P1、P3、P4、および P7 に関しては、既存の技術を利用することができる。ここで、処理 P1~P7 について簡単に説明しておく。運動の計測 (P1)

人間などの身体運動を計測するためには様々な方法が開発されており、例えば、次のような計測方法が挙げられる (持丸正明: 運動計測技術の動向と課題、バイオメカニズム学会誌、Vol.20, No.1, pp.23-32, 1996)。

(a) ゴニオメータを利用する電気角度計測手法

(b) ビデオカメラ等を利用する光学式電子計測手法

8

(c) 磁気センサによる手法

(d) 加速度計や超音波センサによる手法

これらのうちの計測方法を利用するかは、計測対象や利用分野に依存する。しかし、どの方法を用いたとしても、計測される運動データは、関節の角度や身体上の特定部分の位置をある一定のサンプリング間隔で記録した時系列データとなる。

【0024】運動データの符号化 (P2)

計測した運動データをプレイバックすることによってキャラクターに様々な動作を行わせる場合、あらかじめ多数の運動データを計測し、保存しておかなければならない。多数の動作を含んだ運動データを記憶するためには、膨大な記憶容量が必要となる。そこで、本発明では、運動データを符号化することによってデータ圧縮を行い、効率的に運動データを保存できるようにする。これによって、多数の運動データを記録するために必要な記憶容量を削減することができる。

【0025】符号化処理は、時系列として計測された運動データを、ある滑らかさを持つ局所的な基底関数の重み付き加算で近似表現し、その重み係数の配列だけを記録することによって行われる。

【0026】図 3 は、運動データの符号化処理のフローチャートである。データ変換装置は、まず、適当なサンプリング間隔で計測された運動データを入力し (ステップ S1)、滑らかで局所的な基底関数を利用してそれらを近似表現する (ステップ S2)。次に、基底関数の重み係数の配列を生成し (ステップ S3)、それを符号化運動データとして保存して (ステップ S4)、処理を終了する。この符号化運動データは、図 2 の記憶装置 11 に保存される。

【0027】ここで、計測のサンプリング間隔を δt [sec] とし、近似表現を行うための基底関数の時間間隔を $\Delta t = n \delta t$ [sec] とすると、この符号化によって、データの記憶量を n 分の 1 に圧縮することができる。例えば、 $\delta t = 0.01$ [sec]、 $\Delta t = 0.1$ [sec] ならば、データ量は 10 分の 1 に圧縮される。

【0028】図 4 のグラフは、運動データの符号化の例を示している。図 4 において、曲線が計測対象の現実の運動軌跡を表し、曲線上の点が一定のサンプリング間隔で計測された位置データの値を表す。これらの値を時間の指標と共に記録した時系列データが運動データである。また、×印の点が各基底関数の重み係数に対応し、これらの重み係数の配列が符号化運動データとなる。

【0029】符号化運動データの保存と取り出し (P3、P4)

符号化運動データは、局所的な基底関数の重み係数の配列、すなわち数を要素とする配列で表現される。この重み係数の配列をハードディスクなどの記憶装置 11 に保存すること、および、記憶装置 11 から取り出すことは

容易である。

【0030】運動データの編集 (P5)

運動データの再利用性を向上させるため、運動データの編集が実現される。編集は時系列の運動データそのものではなく、符号化運動データに対して行われる。データの再利用性を向上させることは、運動データの記憶容量を減少させる利点も合せ持っている。

【0031】符号化運動データの編集では、例えば、次のような各種機能を設ける。

- (a) 座標変換 (キネマティック変換)
- (b) 時間に関する変換
- (c) 空間に関する変換
- (d) 分割、連結、融合
- (e) 手動による重み係数の調節
- (f) ダイナミクスを利用した変換

これらの編集機能の詳細については後述するが、このような機能を利用して、例えば、幾つかの運動データを材料にして新しい運動データを作り出すことができる。したがって、少ない量の運動データから様々な動作を再現することができるようになる。

【0032】以上のような符号化運動データの編集は、グラフィカルなインタフェースを介して、ユーザとの対話形式でインタラクティブに行われる。このため、編集後のキャラクターの動作が即座に画面上に表示され、ユーザはその仕上がり具合を視覚的に判断しながら、編集作業を進めることができる。

【0033】運動データの復元 (P6)

符号化運動データから時系列形式の運動データへの復元処理は、符号化の逆の処理により行われる。すなわち、上述の基底関数を符号化運動データで重み付き加算すれば、時系列の運動データを復元できる。このとき、符号化運動データは必ずしも編集されている必要はなく、処理P2で得られたデータをそのまま用いてもよい。

【0034】運動の再現 (P7)

運動データを時系列形式に復元した後、キャラクターの3次元キネマティック・モデルにあてはめて関節角度などを設定すれば、運動データをキャラクターに再現させることができる。

【0035】本実施形態のデータ変換装置としては、例えば、図5に示すような情報処理装置 (コンピュータ) が用いられる。図5の情報処理装置は、CPU21、メモリ22、入力装置23、出力装置24、外部記憶装置25、媒体駆動装置26、ネットワーク接続装置27、および計測装置28を備え、それらはバス29により互いに接続されている。

【0036】メモリ22には、処理に用いられるプログラムとデータが格納される。メモリ22としては、例えばROM (read only memory)、RAM (random access memory) 等が用いられる。CPU21は、メモリ22を利用してプログラムを実行することにより、データ変

換装置の各処理を行う。

【0037】入力装置23は、例えば、キーボード、ポインティングデバイス、タッチパネルなどであり、必要な指示や情報の入力に用いられる。また、出力装置24は、例えば、ディスプレイやプリンタ等であり、キャラクターの動作やユーザへの問い合わせ事項を出力する。

【0038】外部記憶装置25は、例えば、磁気ディスク装置、光ディスク装置、光磁気ディスク (magneto-optical disk) 装置などであり、符号化運動データを保存する。この外部記憶装置25に、上述のプログラムとデータを保存しておき、必要に応じて、それらをメモリ22にロードして使用することもできる。外部記憶装置25は、図2の記憶装置11として用いられる。

【0039】媒体駆動装置26は、可搬記録媒体30を駆動し、その記録内容にアクセスする。可搬記録媒体30としては、メモリカード、フロッピーディスク、CD-ROM (compact disk read only memory)、光ディスク、光磁気ディスクなど、任意のコンピュータ読み取り可能な記録媒体が用いられる。この可搬記録媒体30に上述のプログラムとデータを格納しておき、必要に応じて、それらをメモリ22にロードして使用することもできる。

【0040】ネットワーク接続装置27は、LAN (local area network) などの任意のネットワーク (回線) を介して外部の装置と通信する。これにより、必要に応じて、上述のプログラムとデータを外部の装置から受け取り、それらをメモリ22にロードして使用することもできる。また、計測装置28は、例えば、ゴニオメータ、ビデオカメラ、磁気センサ、加速度計、超音波センサなどであり、運動データの計測に用いられる。

【0041】図6は、図5の情報処理装置にプログラムとデータを供給することのできるコンピュータ読み取り可能な記録媒体を示している。可搬記録媒体30や外部のデータベース31に保存されたプログラムとデータは、メモリ22にロードされる。そして、CPU21は、そのデータを用いてそのプログラムを実行し、必要な処理を行う。

【0042】図5の情報処理装置を用いて、図2のような処理を行うデータ変換装置の構成を示すと、図7のようになる。図7において、入力された運動データは、メモリ22上でCPU21の符号化部32により符号化され、符号化運動データに変換されて、記憶装置25に保存される。保存された符号化運動データはメモリ22上に読み出された後、CPU21の編集部33により編集され、ディスプレイ装置24'の画面に表示される。ディスプレイ装置24'は、図5の出力装置24に対応する。

【0043】次に、図8から図22までを参照しながら、図2に示した処理過程の中で、本発明に特に関係する部分、すなわち、運動データの符号化処理と符号化運

11

動データの編集処理の具体例について説明する。

【0044】運動データの符号化では、時系列として計測された運動データを、ある滑らかさを持つ局所的な基底関数の重み付き加算で近似表現し、基底関数の重み係数の配列を符号化運動データとする。

【0045】これまでの人間の運動の計測と解析の研究により、人の腕の運動がある種類の滑らかさの規範を満たすように実行されているとする仮説が提案されている。ここで、滑らかさの規範とは、例えば、jerk最小規範やトルク変化最小規範などを指し、jerkとは加速度の時間微分を指す。したがって、本実施形態の符号化方法のように、運動データを滑らかさを持った基底関数の重み付き加算で近似表現することは妥当であると考えられる。ここでは、このような基底関数の一例として、等間隔の節点を有する正規化されたB-スプライン関数を用いることにする。

【0046】ここで、等間隔の非多重の節点を有する正規化されたB-スプライン関数を計算するためのアルゴリズムについて説明する。このアルゴリズムは、一般的なB-スプライン関数を安定に計算するために、de Boor およびCox によって提案されたアルゴリズム (C. de Boor: A Practical Guide to Splines. Springer-Verlag, 1978) を単純化することにより導出される。

【0047】 p 個の非多重の実数の節点 $s =$

$$B_k(s) = \begin{cases} N_{k-j,k}(s-j), & j \leq s \leq j+1, j=0, \dots, k \\ 0, & s \leq 0, k+1 \leq s \end{cases}$$

【0052】のように計算することができる。ここで、関数 $N_{0,k}(s), \dots, N_{k,k}(s)$ は再帰的に、

$$\begin{cases} N_{0,0}(s) = 1, \\ N_{0,k}(s) = \frac{1-s}{k} N_{0,k-1}(s), \\ N_{j,k}(s) = \frac{k-j+s}{k} N_{j-1,k-1}(s) + \frac{1+j-s}{k} N_{j,k-1}(s), j=1, \dots, k-1 \\ N_{k,k}(s) = \frac{s}{k} N_{k-1,k-1}(s) \end{cases}$$

【0054】のように定義され、関係

【0055】

$$\sum_{j=0}^k N_{j,k}(s) = 1, 0 \leq s \leq 1$$

【0056】を満たす。このとき、等間隔の非多重の節点 $s = 1, \dots, m+k-1$ を有する正規化された k 次のスプライン関数 $S_k(s; 1, \dots, m+k-1)$ は、対応するB-スプライン関数 $B_k(s - (1-k)), \dots, B_k(s - (m+k-2))$ の線形結

$$S_k(s) = \sum_{i=1-k}^{m+k-1} c_i B_k(s - (i-1))$$

【0058】のように記述することができる。(1) 式に示されるB-スプライン関数の局所的な性質により、

12

* s_1, \dots, s_p (ただし、 $s_1 < \dots < s_p$) を有する k 次のスプライン関数 $S_k(s) = S_k(s; s_1, \dots, s_p)$ は、次の2つの条件を満たす実関数として定義される (T.N.E. Greville, Ed.: Theory and Applications of Spline Functions. Academic Press, 1969)。

【0048】第1の条件は、 $i = 0, \dots, p$ に対する各区間 (s_i, s_{i+1}) において、 $S_k(s)$ が s に関する高々 k 次の多項式であることである。また、第2の条件は、区間 $(-\infty, +\infty)$ において $S_k(s)$ は C^{k-1} 級に属すること、すなわち、 $j = 1, \dots, k-1$ に対して $S_k(s)$ の j 階の導関数が存在し、連続であることである。ただし、 $s_0 = -\infty$ および $s_{p+1} = +\infty$ とする。

【0049】B-スプライン関数は局所的かつ最小の台を有する、スプライン関数の基底として定義される。すなわち、任意のスプライン関数は対応するB-スプライン関数の線形結合によって構成することができる。

【0050】de Boor およびCox によるアルゴリズムを変形することによって、等間隔の非多重の節点 $s = 0, \dots, k+1$ を有する正規化された k 次のB-スプライン関数 $B_k(s; 0, \dots, k+1)$ は、

【0051】

【数1】

(1)

(2)

(3)

☆合として、定数 $c_{1-k+1}, \dots, c_{m+k-1}$ を適切に選択することにより、

【0057】

【数4】

(4)

(4) 式は、

【0059】

【数 5】

$$S_k(s) = \sum_{i=l-k+1}^{j+1} c_i B_k(s-(i-1)), \quad j \leq s \leq j+1, \quad j=l, \dots, m+k-2 \quad (5)$$

【0060】のように変形でき、さらに、

* 【数 6】

【0061】

*

$$S_k(s) = \sum_{i=0}^k c_{i+j-k+1} N_{i,k}(s-j), \quad j \leq s \leq j+1, \quad j=l, \dots, m+k-2 \quad (6)$$

【0062】と変形できる。特に、 $c_{l-k+1} = \dots =$ 10 ※ 【0063】 $c_{m+k-1} = 1$ の場合には、(3) 式により、(6) 式

【数 7】

は、

※

$$S_k(s) = \sum_{i=0}^k N_{i,k}(s-j) = 1, \quad j \leq s \leq j+1, \quad j=l, \dots, m+k-2 \quad (7)$$

【0064】となり、それゆえ、

★ 【数 8】

【0065】

★

$$S_k(s) = 1, \quad l \leq s \leq m+k-1 \quad (8)$$

【0066】となる。以下では、まず、このような性質を持つ k 次の B-スプライン関数を用いた任意の関数の近似表現について簡単に説明し、次に、B-スプライン関数の重み係数を利用した運動データの符号化方法を説明する。そして、最後に、具体的な例として、等間隔の正規化された 3 次の B-スプライン関数を用いた場合を示す。

【0067】ある滑らかな関数 $f(t)$ ($0 \leq t \leq T$) を等間隔 Δt の正規化された B-スプライン関数を用い ☆

20 ☆ 近似的に表現することを考える。いま、変数 t を間隔 Δt で離散化し、離散点を $t = i \Delta t$ (i は整数) と記すことにする。

【0068】このとき、関数 $f(t)$ は等間隔 Δt で配置された $(N+k)$ 個の k 次の B-スプライン関数 $B^{(k)}(t)$ を利用して、次式のように近似表現することができる。

【0069】

【数 9】

$$f(t) \approx \sum_{i=-k}^{N-1} w_i B^{(k)}(t-i\Delta t) = \sum_{i=-k}^{N-1} w_i B_i^{(k)}(t), \quad (0 \leq t \leq T) \quad (9)$$

【0070】ただし、 $B^{(k)}(t)$ は k 次の B-スプライン関数である。また、 $B^{(k)}_i(t)$ は $B^{(k)}(t)$ ◆

◆ を $i \Delta t$ だけ平行移動した関数であり、次式で与えられる。

$$B^{(k)}_i(t) = B^{(k)}(t-i\Delta t) \quad (10)$$

また、 N は、

$$N = T / \Delta t \quad (11)$$

を満たすものとし、 w_i ($i = -k, \dots, N-1$) は B-スプライン関数 $B^{(k)}_i(t)$ の重みを表す係数である。

は、いくつかの B-スプライン関数 $B^{(3)}_i(t)$ の線形結合として表される。

【0072】また、関数 $f(t)$ が m 次元の関数である

【0071】図 8 は、位置を表す関数 $x(t)$ を、等間隔の 3 次の B-スプライン関数 $B^{(3)}(t)$ を用いて近似表現した例を示している。この場合、関数 $x(t)$ ◆

40 場合、(9) 式は次のように表現される。

【0073】

【数 10】

$$f(t) \approx \sum_{i=-k}^{N-1} w_i B^{(k)}(t-i\Delta t) = \sum_{i=-k}^{N-1} w_i B_i^{(k)}(t) = Wb^{(k)}(t). \quad (12)$$

【0074】ただし、

$$f(t) = [f_1(t), \dots, f_m(t)]^T \quad (13)$$

($f(t)$ は m 次元の縦ベクトル)

$$b^{(k)}(t) = [B^{(k)}_{-k}(t), \dots, B^{(k)}_{N-1}(t)]^T \quad (14)$$

($b^{(k)}(t)$ は $(N+k)$ 次元の縦ベクトル)

15

$$w_i = [w_{i,1}, \dots, w_{i,m}]^T$$

(w_i は m 次元の縦ベクトル)

$$W = [w_{-k}, \dots, w_{N-1}]$$

(W は m × (N + k) の行列) である。

【0075】本発明の符号化では、計測によって得られた運動データを(12)式のようにB-スプライン関数を用いて近似表現して、B-スプライン関数の重み係数の行列 $W = [w_{-k}, \dots, w_{N-1}]$ を運動データの符号化結果とする。以下に、運動データから $W =$

$[w_{-k}, \dots, w_{N-1}]$ を求めるまでの処理過程を示す。

$$\delta t = \Delta t / n$$

$$T = M \cdot \delta t = N \cdot \Delta t$$

の関係が成り立つとする。サンプリング点の総数は (M + 1) 個である。

【0077】(12)式より、サンプル時刻 $t = i \cdot \delta$ ($i = 0, \dots, M$) での運動 $y(t)$ は、k 次の *

$$y(t) \approx \sum_{i=-k}^{N-1} w_i B_i^{(k)}(t) = W b^{(k)}(t), \quad (t = 0, \delta t, \dots, M\delta t) \quad (19)$$

【0079】したがって、時刻 0 から T までのある 1 つの動作を行ったときに計測された運動データは、★

$$Y = WA$$

(20)

【0081】と近似表現できる。ただし、

【0082】

$$Y = [y(0), \dots, y(M\delta t)] = \begin{bmatrix} y_1(0) & \dots & y_1(M\delta t) \\ \vdots & & \vdots \\ y_m(0) & \dots & y_m(M\delta t) \end{bmatrix}, \quad (21)$$

(m × (M+1) 行列)

$$A = [b^{(k)}(0), \dots, b^{(k)}(M\delta t)] = \begin{bmatrix} B_{-k}^{(k)}(0) & \dots & B_{-k}^{(k)}(M\delta t) \\ \vdots & \ddots & \vdots \\ B_{N-1}^{(k)}(0) & \dots & B_{N-1}^{(k)}(M\delta t) \end{bmatrix}, \quad (22)$$

((N+k) × (M+1) 行列)

【0083】である。行列 W は (16) 式で定義されている。(21) 式の行列 Y は運動データから構成され、

(22) 式の行列 A はサンプリング間隔 δt と B-スプライン関数の間隔 Δt から計算できる。これらの行列 Y、A を (20) 式に代入した後、それを行列 W について解くことは、既存の方法で実行できる。既存の方法と ◆40

$$W = Y A^+ = A^T (A A^T)^{-1}$$

のように求められる。

【0085】図 9 は、(20) 式に基づく運動データの符号化処理の詳細フローチャートである。データ変換装置は、まず、ユーザから指定されたサンプリング間隔 δt と B-スプライン関数の間隔 Δt を設定し (ステップ S11)、それらの値を用いて行列 A の各要素を設定する (ステップ S12)。

【0086】次に、サンプリング間隔 δt で、運動データ $y(0), y(\delta t), y(2\delta t), \dots, y$

16

(15)

(16)

* 【0076】運動データの計測では、関節角度や身体上の特定部位の位置などの m 個の値が計測され、記録されると仮定する。計測対象が行った運動の軌跡を $y(t) \in R^m$ ($0 \leq t \leq T$, t は時刻) とし、この運動をサンプリング間隔 δt で計測した結果を $y(0), y(\delta t), y(2\delta t), \dots, y(M\delta t)$ とする。ただし、

(17)

(18)

* B-スプライン関数を利用して、次式のように近似表現できる。

【0078】

【数11】

★ 【0080】

★ 【数12】

☆ 【数13】

☆

◆しては、例えば、擬似逆行列 (一般化逆行列) を利用する方法やガウス消去法などがある。

【0084】一例として、擬似逆行列を利用する方法を示すと、行列 A の擬似逆行列 $A^+ = A^T (A A^T)^{-1}$ を使用して、行列 W は、

(23)

(Mδt) を計測し (ステップ S13)、(21) 式の行列 Y の各要素を設定する (ステップ S14)。次に、行列 Y、A を (20) 式に代入し (ステップ S15)、それを解いて、行列 $W = [w_{-k}, \dots, w_{N-1}]$ を求める (ステップ S16)。そして、この行列 W を符号化運動データとする。

【0087】ここで、 w_i ($i = -k, \dots, N-1$) の下付添え字 i は運動の時刻を表すのではなく、i 番目の B-スプライン関数 $B^{(k)}_i(t)$ の係数ベクトル

17

ルであることを表している。(17)式が成り立つとき、このような符号化処理により、運動データのデータ量をn分の1に圧縮することができる。

【0088】ところで、本実施形態の符号化処理では、位置や関節角度などの運動データ $y(t)$ を(19)式*

$$\dot{y}(t) \approx \sum_{i=-k}^{N-1} w_i \dot{B}_i^{(k)}(t) = W \dot{b}^{(k)}(t) \quad (24)$$

$$\ddot{y}(t) \approx \sum_{i=-k}^{N-1} w_i \ddot{B}_i^{(k)}(t) = W \ddot{b}^{(k)}(t) \quad (25)$$

【0090】が得られる。したがって、 $y(t)$ が位置データである場合、(24)、(25)式より、速度外1と加速度外2を同一の符号化運動データ $W =$

$[w_{-k}, \dots,$

【0091】

【外1】

$\dot{y}^{(w)}$

【0092】

【外2】

$\ddot{y}^{(w)}$

【0093】 w_{N-1} を用いて表現することができる。つまり、符号化運動データとして行列 W だけを記録しておけば、(19)、(24)、(25)式によって、位置、速度、加速度のすべてを再現することができる。同様に、 $y(t)$ が角度データである場合は、(24)、(25)式によって、符号化運動データ W から角速度、角加速度を再現することができる。

【0094】このように、B-スプライン関数を用いた符号化によれば、B-スプライン関数の時間微分を用い 30 ることで、符号化運動データから、元の運動データのみ*

$$B^{(3)}(t) = \begin{cases} N_{3,3}(t), & 0 \leq t < \Delta t \\ N_{2,3}(t - \Delta t), & \Delta t \leq t < 2\Delta t \\ N_{1,3}(t - 2\Delta t), & 2\Delta t \leq t < 3\Delta t \\ N_{0,3}(t - 3\Delta t), & 3\Delta t \leq t < 4\Delta t \\ 0, & t < 0, 4\Delta t \leq t \end{cases} \quad (26)$$

【0099】ただし、関数 $N_{3-j,3}(t)$ ($j=0, 1, 2, 3$) は以下の通りである。 40 ★ 【0100】

★ 【数16】

$$\begin{cases} N_{0,3}(t) = \frac{1}{6}(1 - (t/\Delta t))^3 \\ N_{1,3}(t) = \frac{1}{6}(4 - 6(t/\Delta t)^2 + 3(t/\Delta t)^3) \\ N_{2,3}(t) = \frac{1}{6}(1 + 3(t/\Delta t) + 3(t/\Delta t)^2 - 3(t/\Delta t)^3) \\ N_{3,3}(t) = \frac{1}{6}(t/\Delta t)^3 \end{cases} \quad (27)$$

【0101】いま、 $t = (j+e)\Delta t$ ($0 \leq e \leq 1$) 形できる。 50 【0102】

18

*で近似表現することを利用している。(19)式を時間微分すると、

【0089】

【数14】

※ならずその時間微分のデータも再現することができる。したがって、少ない記憶容量でより多くのデータを再現できるという利点がある。

【0095】さらに、人間の腕の運動に関する研究によれば、その運動軌跡を等間隔の3次のB-スプライン関数を利用してうまく近似できることが報告されている

(E. Burdet: Algorithms of Human Motor Control and their Implementation in Robotics. PhD thesis No.11

20 692, 外3 (ETH)-Zuerich, 1996)。そこで、次に

【0096】

【外3】

Eidgenössische Technische Hochschule

【0097】、3次のB-スプライン関数を用いた場合について、具体的な符号化の例を示すことにする。等間隔 Δt の非多重の4個の節点 $t=0, \Delta t, 2\Delta t, 3\Delta t$ を持つ正規化された3次のB-スプライン関数 $B^{(3)}(t; 0, \Delta t, 2\Delta t, 3\Delta t)$ は、(2)式より、次のように定義される。

【0098】

【数15】

【数 17】

$$\begin{aligned}
 y((j+e)\Delta t) &= \sum_{i=3}^{N-1} c_i B^{(3)}((j+e-i)\Delta t) \\
 &= c_{j-3} B^{(3)}((e+3)\Delta t) + c_{j-2} B^{(3)}((e+2)\Delta t) + c_{j-1} B^{(3)}((e+1)\Delta t) + c_j B^{(3)}(e\Delta t) \\
 &= c_{j-3} N_{0,3}(e\Delta t) + c_{j-2} N_{1,3}(e\Delta t) + c_{j-1} N_{2,3}(e\Delta t) + c_j N_{3,3}(e\Delta t) \\
 &= c_{j-3} b_0(e) + c_{j-2} b_1(e) + c_{j-1} b_2(e) + c_j b_3(e)
 \end{aligned}
 \tag{28}$$

【0103】ただし、関数 $b_i(s)$ ($i=0, 1, 2, 3, 0 \leq s \leq 1$) は次のように定義される。 * 【0104】

* 10 【数 18】

$$\begin{cases}
 b_0(s) = \frac{1}{6}(1-s)^3 \\
 b_1(s) = \frac{1}{6}(4-6s^2+3s^3) \\
 b_2(s) = \frac{1}{6}(1+3s+3s^2-3s^3) \\
 b_3(s) = \frac{1}{6}s^3
 \end{cases}
 \tag{29}$$

【0105】以上より、等間隔の3次のB-スプライン関数を利用した場合、行列Aは次のように簡単に計算することができる。 * 【0106】

【数 19】

※

$$A = \begin{bmatrix} D & 0 & \cdots & 0 \\ 0 & D & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & D \end{bmatrix},
 \tag{30}$$

$$D = \begin{bmatrix} b_0(0) & b_0(\delta t) & \cdots & b_0(\Delta t) \\ b_1(0) & b_1(\delta t) & \cdots & b_1(\Delta t) \\ b_2(0) & b_2(\delta t) & \cdots & b_2(\Delta t) \\ b_3(0) & b_3(\delta t) & \cdots & b_3(\Delta t) \end{bmatrix}.
 \tag{31}$$

【0107】ただし、 $\Delta t = n \delta t$ である。(31)式の行列Dは、関数 $b_i(s)$ ($i=0, 1, 2, 3, 0 \leq s \leq 1$) と計測サンプリング間隔 δt 、および、B-スプライン関数の間隔 Δt を決定することによって事前に計算することができる。したがって、(30)式の行列Aは、計測した運動データに依存しない定数行列とみなすことができる。

【0108】次に、運動データの編集処理について説明する。運動データの編集が可能になることにより、その再利用性が向上する。編集処理は、運動データそのものに対してではなく、すべて、符号化運動データに対して行われる。したがって、保存してある符号化運動データを時系列の運動データに復元することなく、符号化運動データのままで直接編集を行うことができる。

【0109】本実施形態のデータ変換装置には、

- (a) 座標変換 (キネマティック変換)
- (b) 時間に関する変換

- (c) 空間に関する変換
 - (d) 分割、連結、融合
 - (e) 手動による重み係数の調節
 - (f) ダイナミクスを利用した変換
- などの編集機能が備えられている。

【0110】これらの編集機能を用いたデータ変換の一例を図示すると、図10に示すようになる。例えば、直交座標系における符号化運動データは、(a)の適当な座標変換により、関節座標系における符号化運動データに変換される。また、関節座標系における符号化運動データは、(f)のダイナミクスを利用した変換により、関節トルクを表現する符号化運動データに変換される。そして、これらの符号化運動データは、それぞれ、(b)、(c)、(d)、(e)などの各種の編集処理により、さらに加工することができる。

【0111】図11は、符号化運動データの編集処理のフローチャートである。データ変換装置は、まず、編集

21

対象の符号化運動データを記憶装置から取り出し（ステップS21）、ユーザの指示に従ってそれを編集する（ステップS22）。次に、編集結果をディスプレイ画面に表示して、ユーザによる評価を待つ（ステップS23）。

【0112】編集結果の表示方法には、大きく分けて2つの方法がある。第1の方法は、編集された符号化運動データから運動データを復元して、その時間変化をグラフで表示する方法である。第2の方法は、復元された運動データからキャラクターの動作を再現して表示する方法である。第2の方法は、第1の方法に比べて編集結果が評価しやすいという利点があるが、第1の方法よりも多くの処理を必要とする。どちらの表示方法を選ぶかは、ユーザにより指定される。

【0113】次に、データ変換装置は、同じデータの編集を続けるかどうかをユーザに問合せ（ステップS24）、ユーザが編集続行の指示を入力すると、ステップS22以降の処理を繰り返す。また、ユーザが編集終了の指示を入力すると、次に、他のデータを用いた編集を行うかどうかをユーザに問合わせる（ステップS25）。ここで、ユーザが編集続行の指示を入力すると、ステップS21以降の処理を繰り返し、編集終了の指示を入力すると、処理を終了する。

【0114】次に、上述の各種編集機能の詳細を順に説

$$x = h(q)$$

$$q = h^{-1}(x)$$

【0118】の関係が成り立つ。ここで、(32)式の h は非線形のベクトル関数であり、関節座標 q から直交座標 x への変換を表す。また、(33)式の h^{-1} は、 h の逆変換を表す。このとき、速度に関しては、次式が成

$$\dot{x} = J(q)\dot{q}$$

$$\dot{q} = J^{\#}(q)\dot{x}$$

【0120】ここで、 J は関数 h のヤコビ行列であり、 $J^{\#}$ は広く逆行列を意味する。 $J^{\#}$ としては、外4（以下、 q ドットと記す）と外5（以下、 x ドットと記す）

【0121】

【外4】

\dot{q}

【0122】

【外5】

\dot{x}

【0123】の次元が等しい場合には J^{-1} が使用され、 q ドットの次元が x ドットの次元より大きい場合に

22

*明する。

(a)座標変換（キネマティック変換）

一般に、ある一つの身体運動は直交座標と関節座標など複数の座標系を使って表現することができる。ここで、直交座標は、ある時刻の身体の特定位位（例えば、頭、手先、関節など）の空間内位置によって身体姿勢を表現する座標であり、一方、関節座標は、身体中の関節の角度によって身体姿勢を表現する座標である。

【0115】人間の運動を計測するためによく利用されるモーション・キャプチャー・システムにおいて、ビデオカメラや磁気センサを利用したシステムでは、運動が直交座標で計測され、一方、ゴニオメータを利用したシステムでは、運動が関節座標で計測される。また、運動データを再現する際、コンピュータ内に仮想的に構成した人体モデルを動かすためには、一つの運動を関節座標と直交座標の両方で表現することが有効である。したがって、ある運動データを直交座標と関節座標の両方で表現するため、2つの座標間の座標変換が必要となる。

【0116】ある一つの姿勢が直交座標では $x = [x_1, \dots, x_n]$ と表現され、関節座標では $q = [q_1, \dots, q_m]$ と表現されたとすると、2つの座標の間には、

【0117】

【数20】

(32)

(33)

*り立つ。

【0119】

【数21】

(34)

(35)

は、例えば、擬似逆行列 $J^T (J J^T)^{-1}$ などが使用される。後者の場合の逆行列としては、状況や使用目的に応じて部分的に修正されたいくつかの行列がこれまでに提案されており、それらを利用することができる。

【0124】いま、ある運動を表す符号化運動データが、直交座標では $X = [x_{-k}, \dots, x_{N-1}]$ 、関節座標では $Q = [q_{-k}, \dots, q_{N-1}]$ と表現されたとする。このとき、(24)、(34)、(35)式より、2つの座標で表現した符号化運動データの間には、次式のような関係が成り立つ。

【0125】

【数22】

23

$$\begin{cases} x_i = J(q(i\Delta t))q_i \\ q_i = J'(q(i\Delta t))x_i \end{cases}, (i = -k, \dots, N-1) \quad (36)$$

24

【0126】ここで、(32)、(34)式の x 、 q は、符号化前の運動データを表すのに対して、(36)式の x_i 、 q_i は、符号化運動データを表すことに注意を要する。また、(36)式の $J(q(i\Delta t))$ 、 $J\#(q(i\Delta t))$ は、それぞれ、(34)式の $J(q)$ 、(35)式の $J\#(q)$ に $t = i\Delta t$ を代入して得られる行列である。

(b) 時間に関する変換

時間に関する変換は、運動の開始時刻を遅らせたり、運動時間を伸長または短縮したりする機能を持つ。ここでは、運動開始時刻の遅延演算と運動時間の伸縮演算の2*

$$\begin{aligned} V &= Delay(W, j) \\ &= [\underbrace{w_{-k}, \dots, w_{-k}}_{j+1}, w_{-k+1}, \dots, w_{N-1}] \\ &= [v_{-k}, \dots, v_{N+j-1}] \end{aligned} \quad (37)$$

【0129】すなわち、この演算は、符号化運動データ W の前半に j 個の w_{-k} を新たに付け加えて、運動の初期値に $j\Delta t$ の間だけ留まるようにする変換である。図12は、 $0 \leq t \leq 8$ [sec]の区間における変換前の運動データ（例えば、位置データ）の一例を示している。*

W

$$\begin{aligned} &= [w_{-3}, w_{-2}, w_{-1}, w_0, w_1, w_2, w_3, w_4, w_5, w_6, w_7] \\ &= [0, 0, 0, 1, 2.7, 3.5, 2.7, 1, 0, 0, 0] \quad (38) \end{aligned}$$

図12の区間 $0 \leq t \leq 8$ には、係数 w_0 、 w_1 、 w_2 、 w_3 、 w_4 によりそれぞれ重み付けられたB-スプライン関数 $w_0 B^{(3)}_0(t)$ 、 $w_1 B^{(3)}_1(t)$ 、 $w_2 B^{(3)}_2(t)$ 、 $w_3 B^{(3)}_3(t)$ 、 $w_4 B^{(3)}_4(t)$ の関数形が示されている。

★

*この運動データを、 $\Delta t = 1.0$ 、 $N = 8$ として、3次のB-スプライン関数 $B^{(3)}_i(t)$ ($i = 0, 1, 2, 3, 4$)を用いて符号化すると、次のような符号化運動データが得られる。

【0130】

$$\begin{aligned} V &= [v_{-3}, v_{-2}, v_{-1}, v_0, v_1, v_2, v_3, v_4, v_5, v_6, \\ &\quad v_7, v_8, v_9, v_{10}, v_{11}, v_{12}, v_{13}] \\ &= [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, \\ &\quad 2.7, 3.5, 2.7, 1, 0, 0, 0] \quad (39) \end{aligned}$$

(39)式の符号化運動データにより重み付けされたB-スプライン関数と、それらの総和として得られる運動データを図示すると、図13のようになる。図13では、図12の運動データの区間が $0 \leq t \leq 14$ に拡張され、最初の6秒間は位置が変化していないことが分かる。

☆

★【0131】(38)式の符号化運動データに、開始の時刻を6秒間だけ遅らせる遅延演算 $Delay(W, 6)$ を作用させると、(37)式より、変換後の符号化運動データは次のようになる。

【0132】

☆【0133】また、運動時間の伸縮演算 $TemporalScale(W, \lambda)$ は、運動時間 T を λ 倍に伸縮する変換を表す。ただし、 λ は実数である。演算結果を V とすると、この演算は次のように作用する。

【0134】

【数24】

$$\begin{aligned} V &= TemporalScale(W, \lambda) \\ &= [v_{-k}, \dots, v_{\hat{N}-1}] \end{aligned} \quad (40)$$

【0135】ただし、外6（以下、Nハットと記す）は、
 【0136】
 【外6】

$$\lambda T = \hat{N} \Delta t$$

【0138】を満たし、B-スプライン関数の係数 v_i ($i = -k, \dots, N\text{ハット}-1$)は、

$$\sum_{i=-k}^{\hat{N}-1} v_i B_i^{(k)}(t) = \sum_{i=-k}^{\hat{N}-1} w_i B_i^{(k)}(t/\lambda), (t = j \cdot \delta t, j = 0, \dots, \lambda M) \quad (42)$$

【0140】を満たす。 (42) 式を満たす v_i ($i = -k, \dots, N\text{ハット}-1$)を求めるには、右辺を関数 $y(t)$ とみなして、 (19) 式に基づく図9の符号化処理を実行すればよい。

【0141】例えば、 (38) 式の符号化運動データに★

$$\begin{aligned} V &= [v_{-3}, v_{-2}, v_{-1}, v_0, v_1, v_2, v_3, v_4, v_5, v_6, \\ &\quad v_7, v_8, v_9, v_{10}, v_{11}, v_{12}, v_{13}, v_{14}, v_{15}] \\ &= [0, 0, 0, 0.13, 0.5, 1.09, 1.85, 2.59, \\ &\quad 3.1, 3.3, 3.1, 2.59, 1.85, 1.09, 0.5, \\ &\quad 0.13, 0, 0, 0] \end{aligned} \quad (43)$$

(43) 式の符号化運動データにより重み付けされたB-スプライン関数と、それらの総和として得られる運動データを図示すると、図14のようになる。図14では、図12の運動時間が16秒間に伸張されていることが分かる。

(c) 空間に関する変換

空間に関する変換として、平行移動、回転、および拡大縮小（ただし、運動時間 T は不変）について説明する。30
 これらの変換は、主に直交座標での符号化運動データに☆

$$\begin{aligned} V &= \text{Translate}(W, \delta w) \\ &= [w_{-k} + \delta w, \dots, w_{N-1} + \delta w] \\ &= [v_{-k}, \dots, v_{N-1}] \end{aligned} \quad (44)$$

【0145】例えば、 (38) 式の符号化運動データに対して、運動データを1.0だけ平行移動する演算 $\text{Translate}(W, 2)$ を作用させると、 (44) 式より、変◆

$$\begin{aligned} V &= [v_{-3}, v_{-2}, v_{-1}, v_0, v_1, v_2', v_3, v_4, v_5, v_6, v_7] \\ &= [1, 1, 1, 2, 3.7, 4.5, 3.7, 2, 1, 1, 1] \end{aligned} \quad (45)$$

(45) 式の符号化運動データにより重み付けされたB-スプライン関数と、それらの総和として得られる運動データを図示すると、図15のようになる。図15では、図12の運動データが1.0だけ上方に平行移動していることが分かる。

【0147】また、直交座標の原点に関して角度 q だけ

【0137】
 【数25】

*

(41)

※【0139】
 ※【数26】

★対して、運動時間を2倍に伸張する演算 $\text{TemporalScale}(W, 2)$ を作用させると、 (42) 式より、変換後の符号化運動データは次のようになる。

【0142】

☆対して行われる。いま、符号化運動データ $W =$

$[w_{-k}, \dots, w_{N-1}]$ について、 $w_i \in \mathbb{R}^m$ であるものとする。

【0143】このとき、運動データを $\delta w \in \mathbb{R}^m$ だけ平行移動する演算 $\text{Translate}(W, \delta w)$ によって、符号化運動データ W は次のような符号化運動データ V に変換される。

【0144】
 【数27】

(44)

◆換後の符号化運動データは次のようになる。

【0146】

回転する演算 $\text{Rotation}(W, q)$ によって、符号化運動データ W は次のような符号化運動データ V に変換される。

【0148】
 【数28】

27

$$\begin{aligned} V &= \text{Rotation}(W, q) \\ &= [R(q)w_{-k}, \dots, R(q)w_{N-1}] \\ &= [v_{-k}, \dots, v_{N-1}] \end{aligned}$$

【0149】ただし、 $R(q)$ は、係数ベクトル w_i を原点に関して角度 q だけ回転させる回転行列である。例えば、直交座標が2次元の場合は、次の行列 $R(q)$ が*

$$R(q) = \begin{bmatrix} \cos(q) & \sin(q) \\ -\sin(q) & \cos(q) \end{bmatrix} \quad (47)$$

【0151】また、運動時間を変えないで運動データのサイズだけを λ 倍する演算 $\text{SpatialScale}(W, \lambda)$ によって、符号化運動データ W は次のような符号化運動デ*

$$\begin{aligned} V &= \text{SpatialScale}(W, \lambda) \\ &= [\lambda w_{-k}, \dots, \lambda w_{N-1}] \\ &= [v_{-k}, \dots, v_{N-1}] \end{aligned} \quad (48)$$

【0153】例えば、(38)式の符号化運動データに対して、運動データのサイズを2倍にする演算 $\text{SpatialScale}(W, 2)$ を作用させると、(48)式より、変★

$$\begin{aligned} V &= [v_{-3}, v_{-2}, v_{-1}, v_0, v_1, v_2, v_3, v_4, v_5, v_6, v_7] \\ &= [0, 0, 0, 2, 5.4, 7, 5.4, 2, 0, 0, 0] \end{aligned} \quad (49)$$

(49)式の符号化運動データにより重み付けされたB-スプライン関数と、それらの総和として得られる運動データを図示すると、図16のようになる。図16では、図12の運動データのサイズが全体的に2倍になっていることが分かる。

(d) 分割、連結、融合

次に、1つの符号化運動データを2つに分割したり、逆に複数の符号化運動データから新しい符号化運動データを作り出す編集機能について説明する。ここでは、符号★

$$\text{Cut1}(W, j) = \begin{cases} W^{(1)} = [w_{-k}^{(1)}, \dots, w_j^{(1)}] = [w_{-k}, \dots, w_j] \\ W^{(2)} = [w_{-k}^{(2)}, \dots, w_{-k+N-j-1}^{(2)}] = [w_{j+1}, \dots, w_{N-1}] \end{cases} \quad (50)$$

【0157】ここで、 w_j は第 j 番目のB-スプライン関数 $B^{(k)}_j(t)$ の重み係数に対応している。この分割演算を作用させる運動データの例として、図17に示すようなものを考える。

【0158】図17の運動データは、 $0 \leq t \leq 15$ [sec] の区間で計測されている。この運動データを、△◆

$$\begin{aligned} W &= [w_{-3}, w_{-2}, w_{-1}, w_0, w_1, w_2, w_3, w_4, w_5, w_6, \\ &\quad w_7, w_8, w_9, w_{10}, w_{11}, w_{12}, w_{13}, w_{14}] \\ &= [1, 1, 1, 1, 2, 2, 2, 3, 3, 3, \\ &\quad 3, 2, 2, 2, 2, 2, 2, 2] \end{aligned} \quad (51)$$

図17の区間 $0 \leq t \leq 15$ には、係数 w_{-3} 、 w_{-2} 、

28

*用いられる。

【0150】
【数29】

※一タ V に変換される。

【0152】
【数30】

★換後の符号化運動データは次のようになる。

【0154】

☆化運動データの2種類の分割演算、複数の符号化運動データの連結演算、および融合演算について示す。

【0155】符号化運動データ $W = [w_{-k}, \dots, w_{N-1}]$ を、第 j 列の係数ベクトル w_j と第 $(j+1)$ 列の係数ベクトル w_{j+1} の間で2つの符号化運動データに分割する演算 $\text{Cut1}(W, j)$ によって、符号化運動データ W は次のように変換される。

【0156】
【数31】

40 ◆ $t = 1, 0, N = 15$ として、3次のB-スプライン関数 $B^{(3)}_i(t)$ ($i = -3, \dots, 14$)を用いて符号化すると、次のような符号化運動データが得られる。

【0159】

50 $w_{-1}, w_0, w_1, w_2, w_3, w_4, w_5, w_6, w$

29

7、w8、w9、w10、w11、w12、w13、w14により
それぞれ重み付けられたB-スプライン関数 $w_{-3}B^{(3)}_{-3}(t)$ 、 $w_{-2}B^{(3)}_{-2}(t)$ 、 $w_{-1}B^{(3)}_{-1}(t)$ 、 $w_0B^{(3)}_0(t)$ 、 $w_1B^{(3)}_1(t)$ 、 $w_2B^{(3)}_2(t)$ 、 $w_3B^{(3)}_3(t)$ 、 $w_4B^{(3)}_4(t)$ 、 $w_5B^{(3)}_5(t)$ 、 $w_6B^{(3)}_6(t)$ 、 $w_7B^{(3)}_7(t)$ 、 $w_8B^{(3)}_8(t)$ 、 $w_9B^{(3)}_9(t)$ 、 $w_{10}B^{(3)}_{10}(t)$ 、 $w_{11}B^{(3)}_{11}(t)$ 、 $w_{12}B^{(3)}$

$$W^{(1)} = [w^{(1)}_{-3}, w^{(1)}_{-2}, w^{(1)}_{-1}, w^{(1)}_0, w^{(1)}_1, w^{(1)}_2, w^{(1)}_3, w^{(1)}_4]$$

$$= [1, 1, 1, 1, 2, 2, 2, 3]$$

$$W^{(2)} = [w^{(2)}_{-3}, w^{(2)}_{-2}, w^{(2)}_{-1}, w^{(2)}_0, w^{(2)}_1, w^{(2)}_2, w^{(2)}_3, w^{(2)}_4, w^{(2)}_5, w^{(2)}_6]$$

$$= [3, 3, 3, 2, 2, 2, 2, 2, 2, 2] \quad (52)$$

(52)式の符号化運動データ $W^{(1)}$ により重み付けされたB-スプライン関数と、それらの総和として得られる運動データを図示すると、図18のようになる。また、符号化運動データ $W^{(2)}$ により重み付けされたB-スプライン関数と、それらの総和として得られる運動データを図示すると、図19のようになる。図18および図19の運動データは、図17の運動データを2つに分

* $_{12}(t)$ 、 $w_{13}B^{(3)}_{13}(t)$ 、 $w_{14}B^{(3)}_{14}(t)$ の関数形が示されている。

【0160】(51)式の符号化運動データに、 w_4 と w_5 の間で分割する演算Cut1($W, 4$)を作用させると、(50)式より、変換後の符号化運動データは次のようになる。

【0161】

*割した結果を表している。

【0162】また、符号化運動データ W を時刻 $t = j \Delta t$ で2つの符号化運動データに分割する演算Cut2(W, j)によって、符号化運動データ W は次のように変換される。

【0163】

【数32】

$$Cut2(W, j) = \begin{cases} W^{(1)} = [w^{(1)}_{-k}, \dots, w^{(1)}_{j-1}] = [w_{-k}, \dots, w_{j-1}] \\ W^{(2)} = [w^{(2)}_{-k}, \dots, w^{(2)}_{N-j-1}] = [w_{j-k}, \dots, w_{N-1}] \end{cases} \quad (53)$$

【0164】例えば、(51)式の符号化運動データに、時刻 $t = 7 \Delta t = 7$ で分割する演算Cut2($W, 7$)を作用させると、(53)式より、変換後の符号化

★運動データは次のようになる。

【0165】

$$W^{(1)} = [w^{(1)}_{-3}, w^{(1)}_{-2}, w^{(1)}_{-1}, w^{(1)}_0, w^{(1)}_1, w^{(1)}_2, w^{(1)}_3, w^{(1)}_4, w^{(1)}_5, w^{(1)}_6]$$

$$= [1, 1, 1, 1, 2, 2, 2, 3, 3, 3]$$

$$W^{(2)} = [w^{(2)}_{-3}, w^{(2)}_{-2}, w^{(2)}_{-1}, w^{(2)}_0, w^{(2)}_1, w^{(2)}_2, w^{(2)}_3, w^{(2)}_4, w^{(2)}_5, w^{(2)}_6, w^{(2)}_7]$$

$$= [3, 3, 3, 3, 2, 2, 2, 2, 2, 2, 2] \quad (54)$$

(54)式の符号化運動データ $W^{(1)}$ により重み付けされたB-スプライン関数と、それらの総和として得られる運動データを図示すると、図20のようになる。また、符号化運動データ $W^{(2)}$ により重み付けされたB-スプライン関数と、それらの総和として得られる運動データを図示すると、図21のようになる。図20および図21の運動データは、図17の運動データを2つに分割した結果を表している。

【0166】演算Cut1により得られた図18の運動データと、演算Cut2により得られた図20の運動データ

☆を比較すると、前者は図17の運動データの時刻 $t = 7$ の値を含んでいないのに対して、後者はその値を含んでいることが分かる。このように、演算Cut2は、分割時刻の前後の係数ベクトルを分割後の符号化運動データ $W^{(1)}$ と $W^{(2)}$ の両方に残しているため、元の運動データの分割点付近の形状を保存しながら、符号化運動データを分割することができる。

【0167】また、次のような2つの符号化運動データ

【0168】

【数33】

$$W^{(1)} = [w^{(1)}_{-k}, \dots, w^{(1)}_{N_1-1}], W^{(2)} = [w^{(2)}_{-k}, \dots, w^{(2)}_{N_2-1}] \quad (55)$$

【0169】を連結して1つの符号化運動データを作る 50 演算Connect($W^{(1)}, W^{(2)}$)によって、以下のよう

な新しい符号化運動データ V が作成される。

【0170】

$$V = \text{Connect}(W^{(1)}, W^{(2)}) = [w_{-k}^{(1)}, \dots, w_{N_1-1}^{(1)}, w_{-k}^{(2)}, \dots, w_{N_2-1}^{(2)}] \\ = [v_{-k}, \dots, v_{N_1+N_2+k-1}] \quad (56)$$

【0171】例えば、(52) 式の符号化運動データ $W^{(1)}$ と $W^{(2)}$ に、演算 $\text{Connect}(W^{(1)}, W^{(2)})$ を作
用させると、(56) 式より、(51) 式の符号化運動
データ W が生成される。

【0172】また、(55) 式の 2 つの符号化運動デー

$$V = \text{Fusion}(W^{(1)}, W^{(2)}) = [w_{-k}^{(1)} + w_{-k}^{(2)}, \dots, w_{N_1-1}^{(1)} + w_{N_1-1}^{(2)}] \\ = [v_{-k}, \dots, v_{N-1}] \quad (57)$$

【0174】ただし、(57) 式では $N = \max(N_1, N_2)$ である。 N_1 と N_2 が異なる場合、(57) 式の符号化運動データ $W^{(1)}$ と $W^{(2)}$ のうちで、運動時間が短い方の不足分の係数ベクトルとして 0 ベクトル

※タを融合して (重ね合わせて) 1 つの符号化運動データを作る演算 $\text{Fusion}(W^{(1)}, W^{(2)})$ によって、以下のような新しい符号化運動データ V が作成される。

【0173】

【数35】

★ルを補足して計算する。例えば、 $N_1 = N_2 + j$ のときは、(57) 式は次のようになる。

【0175】

【数36】

$$V = \text{Fusion}(W^{(1)}, W^{(2)}) = [w_{-k}^{(1)} + w_{-k}^{(2)}, \dots, w_{N_2-1}^{(1)} + w_{N_2-1}^{(2)}, \underbrace{w_{N_2}^{(1)}, \dots, w_{N_1-1}^{(1)}}_{\text{★ル}}] \quad (58)$$

【0176】例えば、(52) 式の符号化運動データ $W^{(1)}$ と $W^{(2)}$ に、演算 $\text{Fusion}(W^{(1)}, W^{(2)})$ を作
用させると、(57) 式より、次のような符号化運動デ

☆ータが生成される。

【0177】

$$V = [v_{-3}, v_{-2}, v_{-1}, v_0, v_1, v_2, v_3, v_4, v_5, v_6] \\ = [4, 4, 4, 3, 4, 4, 4, 5, 2, 2] \quad (59)$$

(59) 式の符号化運動データにより重み付けされたベースライン関数と、それらの総和として得られる運動データを図示すると、図22のようになる。図22の運動データは、図18および図19の運動データを融合した結果を表している。

(e) 手動による重み係数の調節

演算 $\text{Modify}(W, j, \delta w)$ は、運動軌跡の部分的な

◆調節や修正を行うため、符号化運動データ $W =$

$[w_{-k}, \dots, w_{N-1}]$ の一つの係数ベクトル w

j ($j = -k, \dots, N-1$) にベクトル δw を加算する演算である。この演算によって、符号化運動データ W は次のように変換される。

【0178】

【数37】

$$\text{Modify}(W, j, \delta w) = [w_{-k}, \dots, \underbrace{w_j + \delta w}_{j\text{-th element}}, \dots, w_{N-1}] \quad (60)$$

【0179】一般に、符号化運動データは滑らかな局所的基底関数であるベースライン関数の係数ベクトルの配列であるので、ある一つの係数ベクトルの値を変化させても、その影響は局所的に留まる。また、全体の運動軌跡は常に滑らかなままである。

(f) ダイナミクスを利用した変換

逆ダイナミクスを利用した変換は、位置、速度、および加速度などの運動を表現する符号化運動データを、関節トルクを表現する符号化運動データへ変換する。逆に、順ダイナミクスを利用した変換は、関節トルクを表現する符号化運動データを、運動を表現する符号化運動データに変換する。

【0180】ところで、人間の腕の運動軌跡が、運動を実行する関節トルクの変化が最小となるように計画されているとする学説がある (Y. Uno, M. Kawato, and R. Suz

uki: Formation and Control of Optimal Trajectory in Human Multijoint Arm Movement. Minimum Torque-Change Model. Biol. Cybern. Vol.61, pp.89-101, 1989)。

【0181】この学説は、腕の運動を引き起こしている関節トルクの時間波形が滑らかであることを主張している。したがって、運動を表現する符号化運動データを関節トルクの符号化運動データに変換し、その後、関節トルクの符号化運動データに対して、上述したような各種の滑らかな編集処理を実行することにより、運動の自然さを保ったままの編集が可能になる。ここで、滑らかな編集処理とは、編集後の運動データが滑らかであることを意味する。

【0182】いま、運動を表現する符号化運動データを $W = [w_{-k}, \dots, w_{N-1}]$ 、関節トルクを表現する

* 【数34】

*

10

★

30

40

50

33

符号化運動データを $U = [u_k, \dots, u_{N-1}]$ とする。このとき、 W を U に変換する逆ダイナミクスの演算 *

$$U = \text{InverseDynamics} (W)$$

また、 U を W に変換する順ダイナミクスの演算 $\text{ForwardDynamics} (U)$ は、次のように記述される。 ※

$$W = \text{ForwardDynamics} (U)$$

以上説明したような各種編集機能を用いた作業は、グラフィカル・ユーザ・インタフェース (GUI) を利用して行われる。すなわち、編集対象の符号化運動データの指定や編集機能の選択および値の設定などは、画面上のボタンなどで表現される編集機能をマウスなどのポインティングデバイスを用いて指示することにより行われる。そして、ディスプレイ装置は、編集中および編集結果の運動を画面上に表示し、常に、編集状況をユーザにわかりやすく伝達する。

【0185】本実施形態では、運動データを近似するための基底関数として、現時点で最適と考えられる B-スプライン関数を用いているが、一般には、他の任意の関数を用いることができる。例えば、基底関数として RBF (radical base function) を用いてもよい。また、人物や動物などの生物だけでなく、自動車、航空機、ロボットなどの任意の動く物体を、運動データの計測対象とすることができる。

【0186】

【発明の効果】本発明によれば、時系列データとして与えられる運動データを符号化することで、運動データを効率的に保存することが可能になる。また、符号化された運動データに対して様々な編集を行い、与えられた運動データとは異なる新しい運動データを生成することもできる。したがって、運動データの再利用性が向上する。

【図面の簡単な説明】

【図1】本発明のデータ変換装置の原理図である。

【図2】運動データの処理過程を示す図である。

【図3】符号化処理の概略フローチャートである。

【図4】第1の運動データを示す図である。

【図5】情報処理装置の構成図である。

【図6】記録媒体を示す図である。

【図7】データ変換装置の構成図である。

【図8】等間隔の3次のB-スプライン関数を示す図である。

【図9】符号化処理の詳細フローチャートである。

【図10】符号化運動データの変換を示す図である。

【図11】編集処理のフローチャートである。

34

* InverseDynamics (W) は、次のように記述される。

$$[0183]$$

(61)

$$[0184]$$

(62)

【図12】第2の運動データを示す図である。

【図13】遅延された運動データを示す図である。

【図14】伸張された運動データを示す図である。

【図15】平行移動後の運動データを示す図である。

【図16】拡大された運動データを示す図である。

【図17】第3の運動データを示す図である。

【図18】第1の分割された運動データを示す図である。

【図19】第2の分割された運動データを示す図である。

【図20】第3の分割された運動データを示す図である。

【図21】第4の分割された運動データを示す図である。

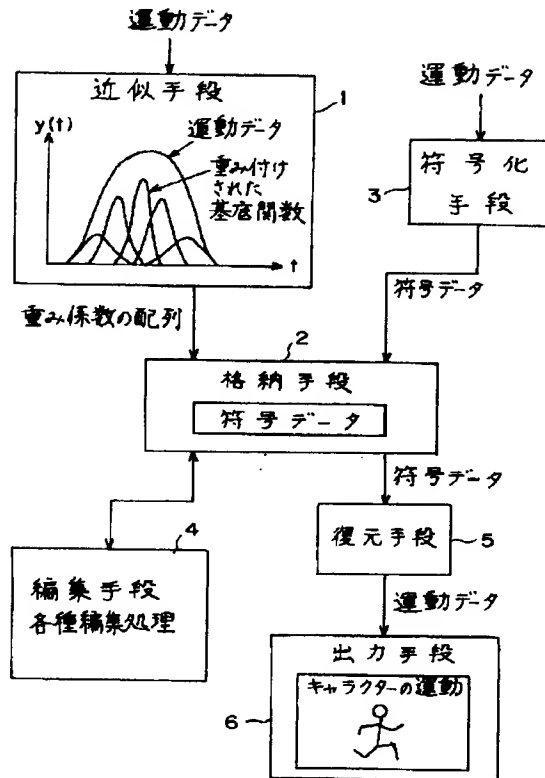
【図22】融合した運動データを示す図である。

【符号の説明】

- 1 近似手段
- 2 格納手段
- 3 符号化手段
- 4 編集手段
- 5 復元手段
- 6 出力手段
- 11 記憶装置
- 21 CPU
- 22 メモリ
- 23 入力装置
- 24 出力装置
- 24' ディスプレイ装置
- 25 外部記憶装置
- 26 媒体駆動装置
- 27 ネットワーク接続装置
- 28 計測装置
- 29 バス
- 30 可搬記録媒体
- 31 データベース
- 32 符号化部
- 33 編集部

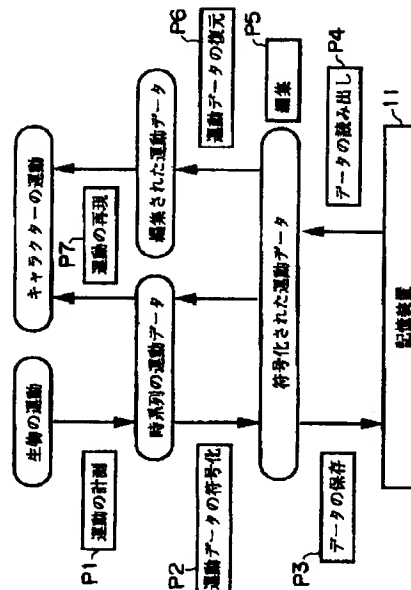
【図1】

本発明の原理図



【図2】

運動データの処理過程を示す図



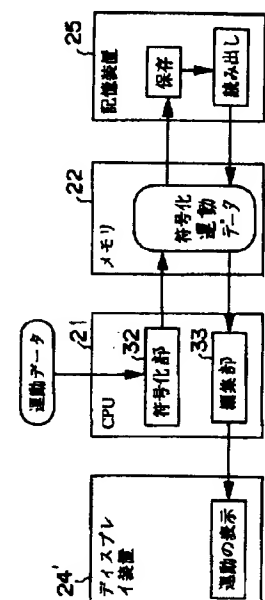
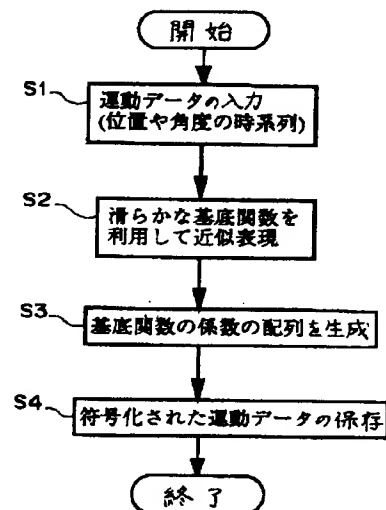
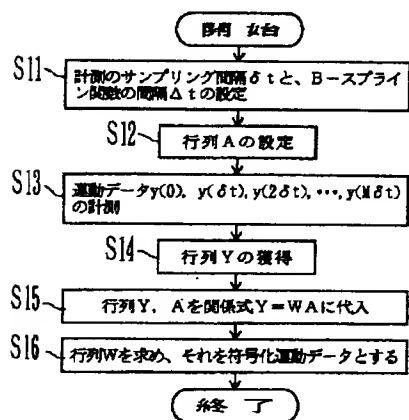
【図3】

【図7】

符号化処理の概略フローチャート データ変換装置の構成図

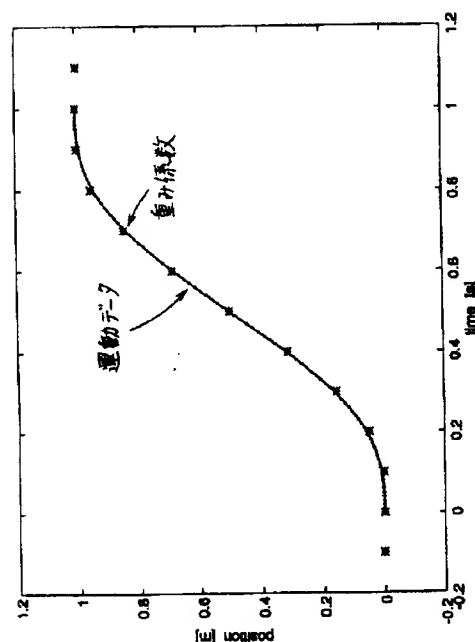
【図9】

符号化処理の詳細フローチャート



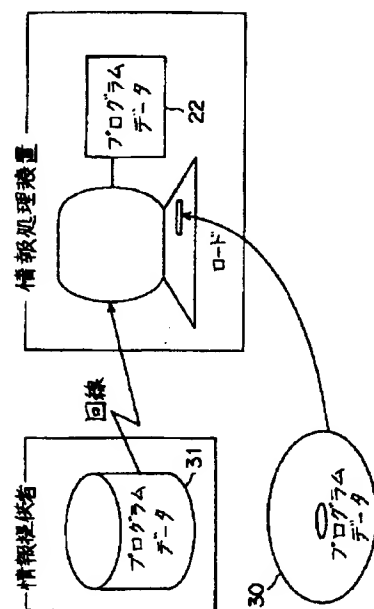
【図 4】

第1の運動データを示す図



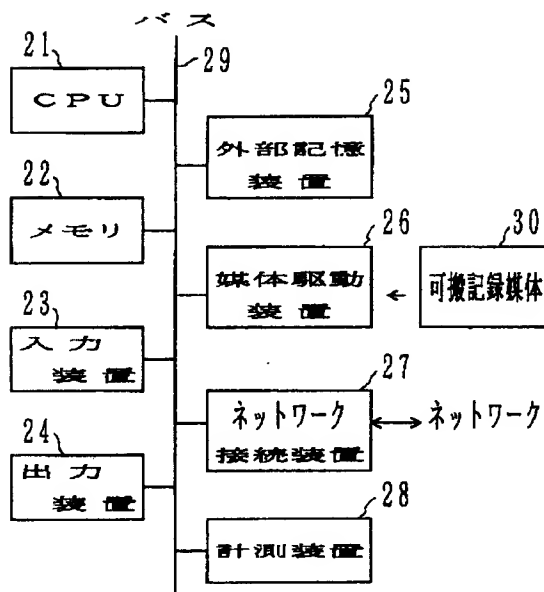
【図 6】

記録媒体を示す図



【図 5】

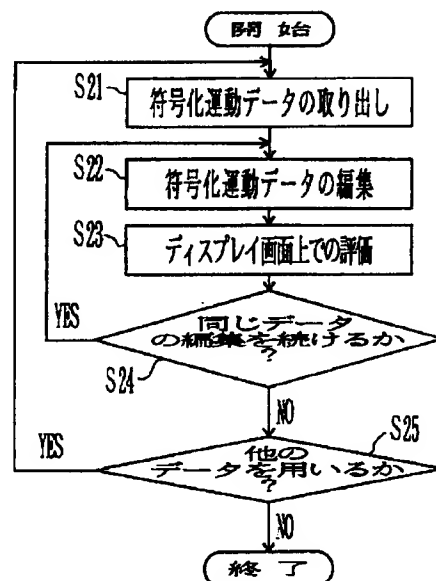
情報処理装置の構成図



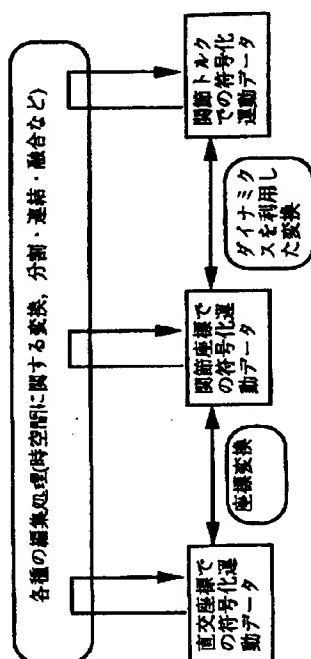
【図 11】

【図 10】

編集処理のフローチャート

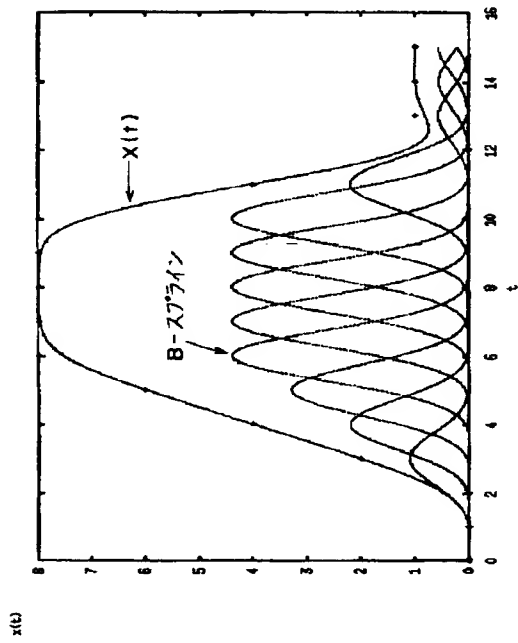


符号化運動データの変換を示す図



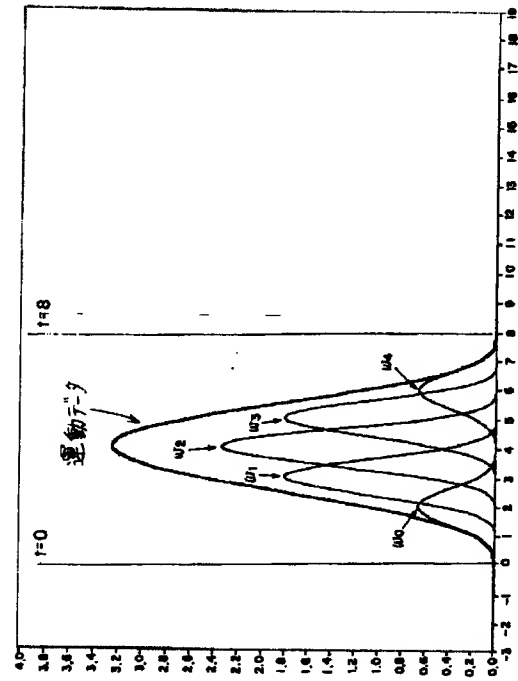
【図 8】

等間隔の 3 次 B-スプライン関数を示す図



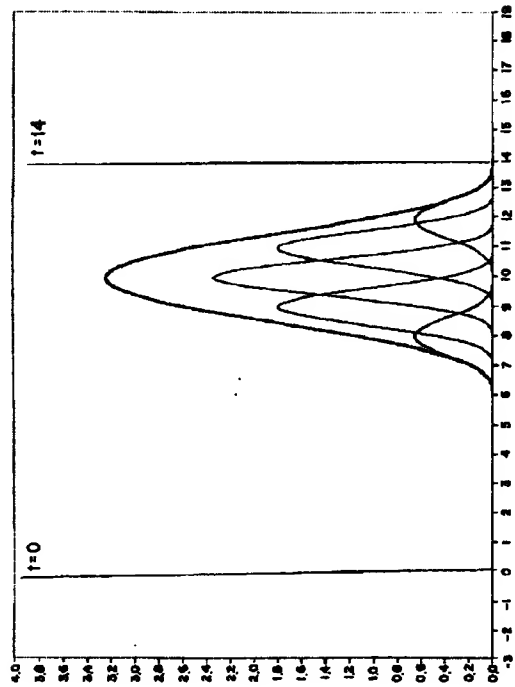
【図 12】

第2の運動データを示す図



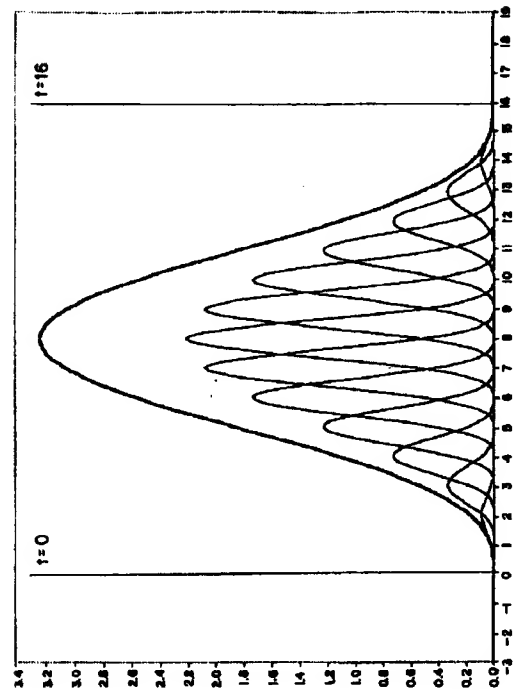
【図 13】

遅延された運動データを示す図



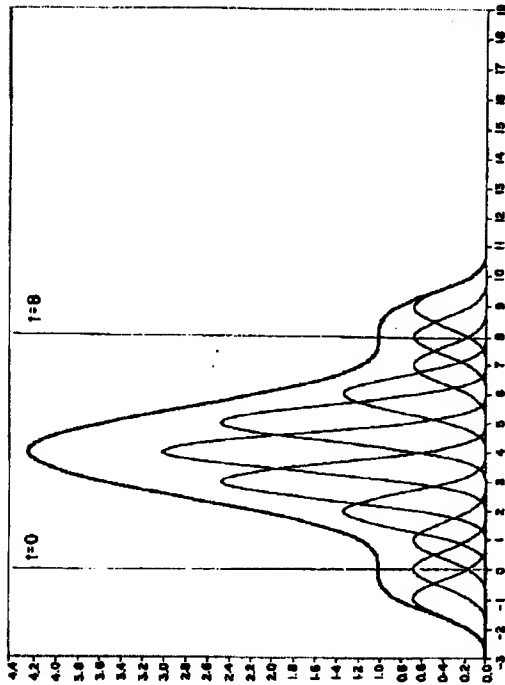
【図 14】

伸長された運動データを示す図



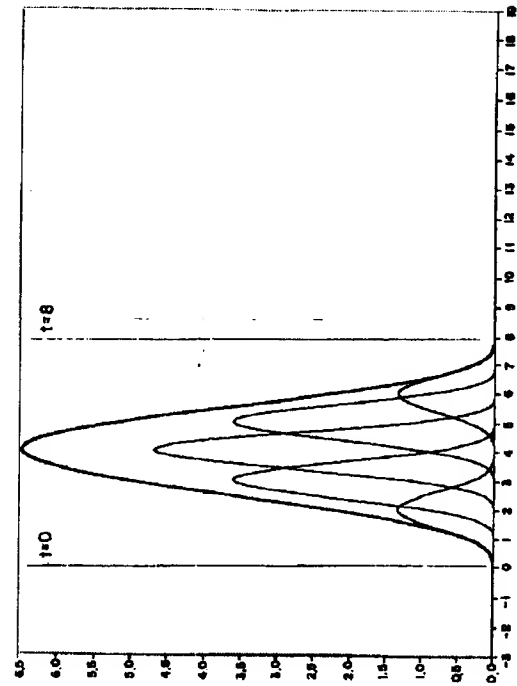
【図 1 5】

平行移動後の運動データを示す図



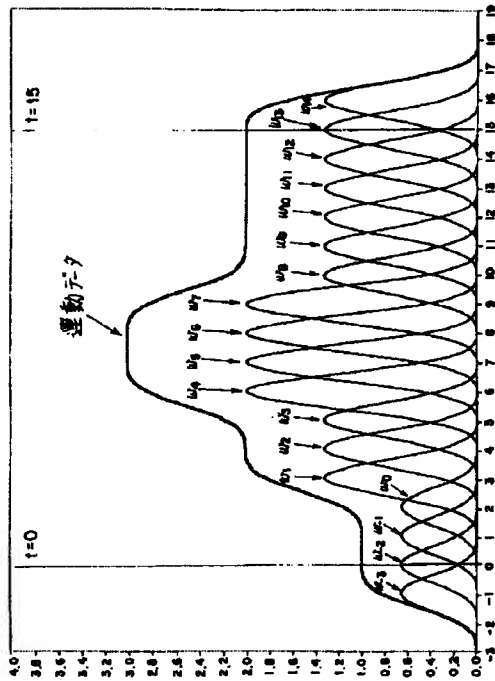
【図 1 6】

拡大された運動データを示す図



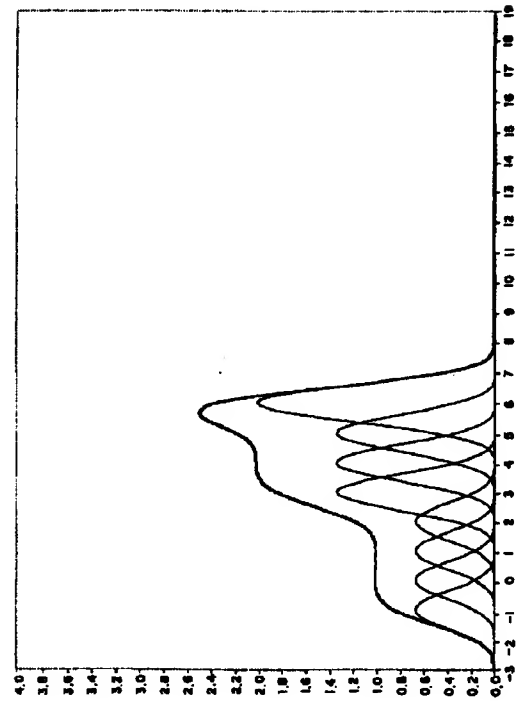
【図 17】

第3の運動データを示す図



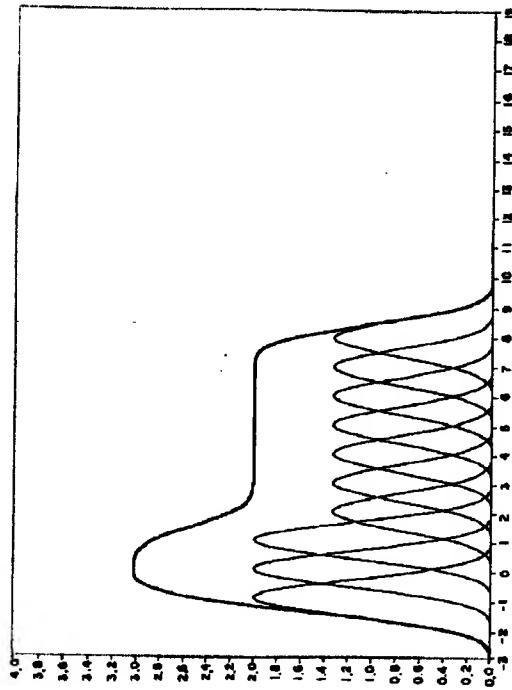
【図 18】

第1の分割された運動データを示す図



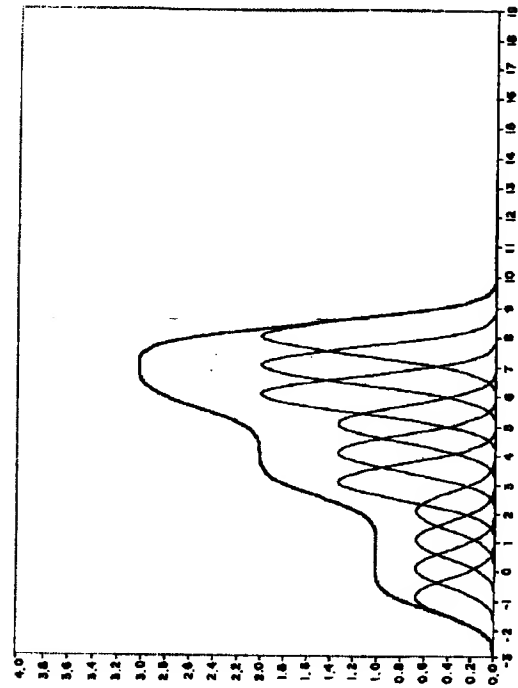
【図 1 9】

第2の分割された運動データを示す図



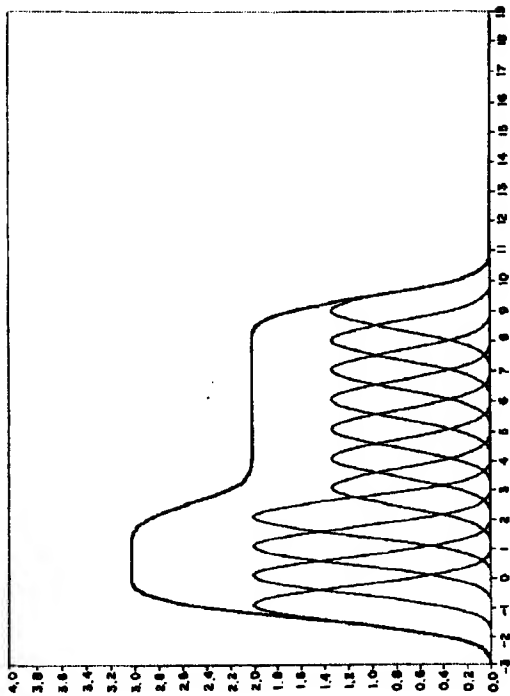
【図 2 0】

第3の分割された運動データを示す図



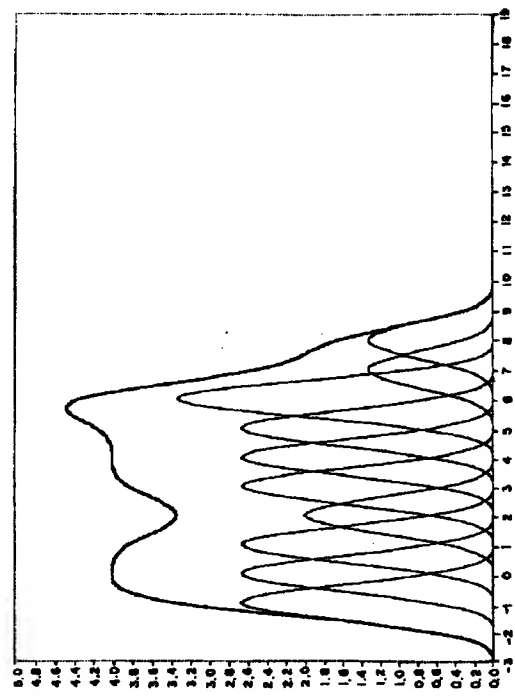
【図 21】

第4の分割された運動データを示す図



【図 22】

融合した運動データを示す図



フロントページの続き

(72)発明者 高山 訓治
 神奈川県川崎市中原区上小田中4丁目1番
 1号 富士通株式会社内

(72)発明者 細木 信也
 神奈川県川崎市中原区上小田中4丁目1番
 1号 富士通株式会社内